APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTORS: Nam-Jin KIM and Hyung-Soo PARK

TITLE: APPARATUS AND METHOD FOR DATABASE SYNCHRONIZATION IN A DUPLEX SYSTEM

ATTORNEYS & ADDRESS: FLESHNER & KIM, LLP
P. O. Box 221200
Chantilly, VA 20153-1200

DOCKET NO.: P-241

APPARATUS AND METHOD FOR DATABASE SYNCHRONIZATION IN A DUPLEX SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

[1] The present invention relates to a duplex system using a main memory database, and particularly, to a method and apparatus for database synchronization in the duplex system.

2. Background of the Related Art

[2] Systems which are required to process, in real-time, a large amount of data stored on an online network, such as an intelligent network supplying an intelligent information service, a home location register (HLR) in a mobile communication system, and a computer system in a bank, are adopting the duplex storage system to process data smoothly and to prevent an overload. The stored data is constructed as a database structure in order to real-time process the large amount of data promptly and the database is stored in the main memory.

[3] When a duplex database system is used, database synchronization between the duplex database systems should be maintained synchronization both when the systems operate properly and after a fault has occurred in one system, once the faulty system is restored. Figure 1 is a block diagram showing a structure of an apparatus for database synchronization in a related art duplex system. Two independent database systems, that is, a local system (primary system) 10 and a remote system (secondary system) 20, form the duplex system. Both of the independent systems have the same structure because they are constructed as duplex systems.

A description of the local system 10 will now be provided that applies equally well to the remote system. The local system 10 includes a database 11, a Database Management System (DBMS) 12, an Application Process (AP) 13, a Database (DB) initialization library 14, an update queue 15, a transmission process 16, a receive process 17, and an arbiter 18.

[4] The database 11 stores the data processed online by the local system 10, the DBMS 12 manages the database 11, and the application process 13 performs query operations to the database 11. The DB initialization library 14 deletes all data stored in database 21 of the remote system 20 when the remote system 20 is rerun, due to a fault occurring in the remote system 20. The update queue 15 stores updated data when components of the database are updated. Transmission process 16 transmits a database deletion message to the remote system 20, under the control of the DB initialization library 14, and transmits data stored in the update queue 15 to the remote system 20. Receive process 17 receives the messages transmitted from the remote system 20 and transmits then to the application process 13. When data stored in the update queue 15 fails to be reflected in the remote system 20, the arbiter 18 records the data in a file.

[5] The operation of the related art duplex database synchronization system is described in relation to its normal operation and to when a synchronization fault occurs.

[6] (A) Normal Operation

[7] When a transaction query process occurs within the local system 10 requiring database synchronization between the duplex systems, application process 13 accesses the database 11 through DBMS 12 and processes the query. Then, the AP 13 stores the transaction

information in the update queue 15 so that the transaction information may be reflected to the database of the remote system 20.

[8] Transmission process 16 checks the update queue 15 periodically and, if there is an update transaction in update queue 15, the transmission process 16 transmits the transaction to receive process 27 of the remote system 20. Receive process 27 of the remote system 20 transmits the transaction information to AP 23, which stores the information to database 21 of the remote system 20. After that, if the AP 23 stores the update transaction successfully, transmission process 26 informs receive process 17, of the local system 10, of its success.

[9] When the transaction update of database 21 is successfully reflected in remote system 20, the transaction information is deleted from the update queue 15 of the local system 10. However, if the transaction update to database 21 fails to be successfully reflected in the remote system, arbiter 18 of the local system 10 generates a log file corresponding to the transaction which failed the synchronization. The log file is processed when the system is restarted.

[10] (B) Synchronization Fault

[11] If a fault occurs in a database system, then the system experiencing the fault is restored by copying the entire contents of the operational system to the faulty system. For example, if a fault is generated in the local system 10, the local system 10 is reinitialized using the operational database of the remote system 20, after the fault is repaired. To accomplished this, the reinitialized local system 10 informs the remote system of the reinitialization and then the DB initialization library 24 of the remote system 20 orders the deletion of all data stored in

database 11 of the local system 10. The local system 10 deletes all data stored in database 11, in accordance with the database initialization order. Remote system 20 copies the data stored in its database 21 to database 11 of the local system 10, once it learns that database 11 of the local system 10 is reinitialized.

[12] During the copying process, messages that are processed by the remote system concurrently with the copying process are stored in update queue 25. After the copying process to the local system 10 is finished, the remote system 20 transmits the data stored in the update queue 25 to the local system 10 and database synchronization between the remote system 20 and the local system 10 is performed for the messages transferred from update queue 25.

[13] Therefore, when a fault occurring in one system is restored, the data stored in its database is entirely deleted. Thereafter, the data stored in the other system is copied to the restored system. Additionally, database transactions occurring in the operational database during the database copying operation are later reflected to the restored database by the update queue.

[14] As described above, the related art apparatus for database synchronization has the problem that the database synchronization process is complex. When a fault is generated is one system of the duplex systems, the faulty system is reinitialized and data stored in the database of the normal system are copied to the database of the reinitialized system, after all the data stored in the faulty database are deleted. Because a large amount of data must be transferred from one database to the other when a fault occurs, a large processing load is placed on the duplex system.

[15] Additionally, when the updated contents of one database are stored in the update queue for reflection to the other system, while the entire content of a database is being copied, the disk approaching time and data transmission time is long. Therefore, the performance of the main memory database may be lowered.

[16] Also, in the event the reflection of the updated data to the other system fails, the failed operation is stored as a log file and the log file is processed when the system is restarted, later. This, too, reduces the operational speed, due to the time required to generate the file input/output and because the file may be damaged.

SUMMARY OF THE INVENTION

[17] An object of the invention is to solve at least the above problems and/or disadvantages and to provide at least the advantages described hereinafter.

[18] It is an object of the present invention to provide a method and an apparatus for database synchronization in duplex systems by which the database synchronization is secured without the lowering of system function, in duplex systems using main memory databases of the same structure.

[19] It is an another object of the present invention to provide a method and an apparatus for database synchronization in duplex systems by which real-time database synchronization is maintained in duplex systems having main memory databases, by using a change list table (CLT) recording a transaction list on the main memory.

[20] To achieve these and other advantages, there is provided an apparatus for database synchronization in duplex systems comprising a database having a CLT for transaction synchronization; a DBMS for managing the database; an AP for recording information about a transaction on the CLT when a transaction needing synchronization is performed, wherein database synchronization is maintained by performing transaction synchronization between the duplex systems using the CLT; a Process Management Block (PRMB) for managing the AP and managing the communications with the other system of the duplex systems; a message transmission process for transmitting the message transmitted from the AP to the other system; and a message receive process for receiving the message transmitted from the other system and transmitting the message to the AP.

[21] To achieve the objects of the present invention, there is provided a method for database synchronization in the duplex systems comprising a process of identifying whether the duplex systems, having the main memory databases, are operating normally; a process of checking whether the previous transaction in the CLT list corresponds to the current transaction; and a process of maintaining database synchronization by reflecting a result of a transaction to the database of the other system, in accordance with the kind of transaction operation performed and the kind of the previous transaction operation existing in the CLT.

[22] To achieve the objects of the present invention, there is provided a method for database synchronization in duplex systems comprising a process of searching the CLT of the normal system database, upon restarting the system in which a fault is generated, and reflecting the updated contents after the fault is generated to the database of the restarted system; and a

process of deleting the information about the updated contents searched in the CLT if the reflection of the updated contents succeeds, and maintaining the information about the updated contents searched in the CLT if the reflection of the contents fails.

[23] The objects of the present invention can be achieved in whole or in part by an improved duplex system having a primary system with a primary database and a redundant system with redundant database, wherein the improvement comprises a Change List Table (CLT) that records key information regarding each transaction performed on the primary database, which must also be performed on the redundant database to maintain synchronization between the primary and redundant databases. The key information for an i^{th} transaction recorded in the CLT is removed from the CLT when the i^{th} transaction is successfully performed on the redundant database.

[24] Additional advantages, objects, and features of the invention will be set forth in part in the description which follows and in part will become apparent to those having ordinary skill in the art upon examination of the following or may be learned from practice of the invention. The objects and advantages of the invention may be realized and attained as particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[25] The invention will be described in detail with reference to the following drawings in which like reference numerals refer to like elements wherein:

[26] Figure 1 illustrates a block diagram showing an embodiment of an apparatus for database synchronization in a related art duplex system;

[27] Figure 2 illustrates a block diagram showing a structure of an apparatus for database synchronization in a duplex system according to a preferred embodiment of the present invention;

[28] Figure 3 illustrates a flow chart showing a method for database synchronization between the duplex system, while the system is operating normally, according to a preferred embodiment of the present invention;

[29] Figure 4 illustrates a flow chart showing a preferred embodiment method for synchronizing a transaction process performed based on the operation of a previous transaction recorded in the CLT, when the current operation is an 'update' transaction;

[30] Figure 5 illustrates a flow chart showing a preferred embodiment method for synchronizing a transaction process performed based on the operation of the previous transaction recorded in the CLT, when the current operation is an 'insertion' transaction;

[31] Figure 6 illustrates a flow chart showing a preferred embodiment method for synchronizing a transaction process performed based on the operation of the previous transaction recorded in the CLT, when the current operation is a 'deletion' transaction;

[32] Figure 7 illustrates a block diagram of a preferred embodiment showing the structure of an apparatus for database synchronization between a system having a fault and a normal system; and

[33] Figures 8A and 8B illustrate flow charts showing a preferred embodiment method of restoring the system of Figure 7 having the fault so that the database synchronization with the normal system can be achieved.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[34] In a preferred embodiment of the invention, the duplex systems having main memory databases of the same structure communicate with each other through a Transmission Control Protocol/Internet Protocol (TCP/IP) socket. In addition, the duplex systems are managed by a Change List Table (CLT) so that if a transaction is changed in one system, of the duplex systems, the update transaction can be reflected to the database of the other system. Also, the present invention is based on an Active-Active (load sharing) operational method of duplex systems but is able to support an Active-Standby operational method as well.

[35] Figure 2 is a block diagram showing the structure of the apparatus for database synchronization in the duplex systems according to a preferred embodiment the present invention. As shown, the duplex systems have the same structure and, therefore, only the structure of the local system (primary system) 100 will be described. The local system includes a database 110, a Database Management System (DBMS) 120, an Application Process (AP) 130, a Process Management Block (PRMB) 140, a message transmission process 150, and a message receive process 160.

[36] Database 110 stores a data table, which is real-time processed by the local system 100, on an online network. The database 110 includes a User Profile Table (UPT) 111, for

storing the user profile, and a CLT 112 for recording the update contents of the UPT 111. The CLT 112 is used to synchronize the databases after a transaction is processed. The DBMS 120 manages the database 110.

[37] AP 130 receives a message to be processed, performs the process transaction by approaching the database 110, and performs database synchronization using the CLT 112 when a transaction requiring synchronization is processed. AP 130 includes an Application Service Element (ASE) 131, a first query interface library 132 interfacing the ASE 131 to the database 110, a second query interface library 133 interfacing the ASE 131 to approach to the message transmission process 150, and a recovery process 134, driven when a fault is generated in a remote system 200.

[38] The PRMB 140 manages the AP 130 and is in charge of communication between the duplex systems. The message transmission process 150 transmits a message, received from the AP 130, to the remote system 200 by the TCP/IP method. In addition, the message receive process 160 receives the message transferred from the remote system 200 using the TCP/IP method.

[39] Also, as shown in Figure 2, a counterpart system 300 communicates with both systems 100, 200 requires the duplex systems 100 and 200 to process its messages as described herein. The operation of the apparatus for synchronizing the database in the duplex systems will now be described for the case of normal operation and the case where a fault has occurred in one system.

[40] (A) Normal Operation.

[41] Both duplex systems 100, 200 receive and process messages simultaneously when the systems are operated according to the Active-Active method. When operated in the Active-Standby method, only one of the duplex systems 100, 200 receives and processes the messages and then subsequently reflects the transaction and then the result to the other system.

[42] With the Active-Standby method, if an operation is performed that may change the status of the database, such as an insert, delete, or update operation, the active local system processes the transaction and transmits a transaction message to the remote system. The transaction message instructs the standby system to execute the transaction in its database.

[43] Figure 3 is a flow chart showing the method for synchronizing the databases of the duplex systems during their normal operation, according to a preferred embodiment of the present invention. When a message is transferred from the counterpart system 300, the PRMB 140 of the local system 100 transmits the message to the AP 130. The AP 130 identifies the query request and performs the required transaction by accessing database 110 (S11 and S12). That is, the ASE 131 of AP 130 accesses the database through the first query interface library 132 and performs the required transaction in the data table of the database 110. If the required transaction is performed successfully, the ASE 131 identifies whether the transaction needs to be synchronized with the remote system 200. If the contents of database 110 (S13) are changed by the recent transaction, then synchronization is required. If the operation of the performed transaction does not need synchronization, the operation is ended. When synchronization is required, the ASE 131 accesses the CLT 112 of database 110 through the first query interface library 132 and determines whether there is a previous transaction corresponding to the current

transaction in the CLT list 112 (S14). If there is not a previous transaction listed in the CLT 112, the ASE 131 accesses the CLT 112 through the first query interface library 132 and records the current transaction to the list. For example, the operation code of the current transaction, index, and key value of the table in which the transaction is generated may be listed in the CLT 112 (S15). If current transaction is recorded in the CLT 112 normally, the ASE 131 transmits a transaction request message to the message transmission process 150, through the second query interface library 133, to reflect the transaction to the remote system 200. Message transmission process 150 transmits the transaction request message to the remote system 200 using a TCP/TP method (S16).

[44] The message receive process 260 of the remote system 200 transmits the transaction request message to the ASE 231 of AP 230 and ASE 231 determines whether the transaction should occur. If so, ASE 231 performs the transaction by accessing database 210. If this corresponding transaction is performed successfully, ASE 231 of the remote system 200 informs the local system 100 of its success through message transmission process 250. When the success of the reflected transaction is reported by the remote system 200 (S19), the ASE 131 of the local system 100 accesses the CLT 112 through the first query interface library 132 and deletes the corresponding transaction list from the CLT 112 (S20).

[45] As described above, if a transaction that needs synchronization happens in one system, the performed transaction is recorded in the CLT of the system. The transaction is deleted from the CLT after the transaction is successfully reflected to the other system and database synchronization is achieved.

[46] However, if a previous transaction list related to the current transaction exists in CLT 112, the ASE 131 identifies the operation type of the current transaction and that of the previous transaction stored in the CLT list (S17). The local system 100 performs a particular synchronization transaction process with the remote system 200 based on the current transaction type and the previous transaction type listed in the CLT. The method for performing the synchronization process will now be described with reference to Figures 4, 5, and 6.

[47] Figure 4 is a flow chart showing the method for performing the synchronization process, based on the operation of the transaction recorded on the CLT, when the current transaction is an 'update' operation. Figure 5 is a flow chart showing the method for performing the synchronization process when the current transaction is an 'insert' operation. Figure 6 is a flow chart showing the method for performing the synchronization process when the current transaction is a 'delete' operation.

[48] As shown in Figure 4, if the current transaction is an 'update' operation (S31) and the previous transaction in the CLT 112 is an 'update' operation, then the update transaction generated in the local system, previously, was not successfully reflected to the remote system 200. Therefore the local system 100 should now reflect the update transaction to the remote system 200 (S32 and S33). If the operation of the current transaction is an 'update' operation (S31) and the operation of the previous transaction listed in the CLT 112 is an 'insert' operation, then an insert transaction happened previously in the local system 100 but was not successfully reflected to the remote system 200. Therefore, the local system 100 should now reflect the 'insert' transaction to the remote system 200 (S32 and S34). If the operation of the transaction

listed in the CLT is a 'delete' operation, an update transaction for the Tuple deleted from the local system 100 is not generated. In other words, no attempt will be made to reflect the 'update' transaction to the remote system 200 for the deleted Tuple.

[49] As shown in Figure 5, if the current transaction is an 'insert' operation (S41) and the operation of the previous transaction listed in the CLT 112 is a 'delete' or 'update' operation, then the previous transaction generated by the local system 100 was not successfully reflected to the remote system 200. Therefore, the local system 100 should now reflect the current insert transaction to the remote system 200 instead of the update transaction (S42 and S44). However, if the previous transaction listed in the CLT 112 is an 'insert' operation, then the insert transaction previously generated by the local system 100 has not been successfully reflected to the remote system 200. Therefore, the insert transaction previously generated is not reflected to the remote system 200 but is ignored. However, the current transaction is reflected to remote system 200 (S43).

[50] Referring now to Figure 6, if the operation of the current transaction is a 'delete' operation (S51) and the previous transaction listed in the CLT 112 is an 'update' operation, then the update transaction generated before was not successfully reflected to the remote system 200. Therefore, the local system 100 should now reflect the current delete transaction to the remote system 200 (S52 and S53). If the operation of the transaction listed in the CLT 112 is a 'delete' operation, the previous delete transaction for the Tuple deleted from the local system 100 is not generated for reflection to the remote system 200. Similarly, when the current transaction is an

'insert' operation and the listed transaction in the CLT 112 is a 'delete' operation, neither transaction gets reflected to the remote system 200.

[51] When ASE 131 reflects the result of a transaction to the remote system successfully, the corresponding transaction is deleted from the CLT 112 list (S19 and S20). However, if the result of the transaction is not reflected to the remote system 200 successfully, ASE 131 maintains the corresponding transaction in the CLT 112 list. Also, when a transaction requiring synchronization is processed successfully and recorded in the CLT but the transaction is not processed successfully in the remote system, then the next transaction reflected to the remote system is determined according to the types of the current and the previously listed CLT operations. Thereby, database synchronization between the local system and the remote system is maintained.

[52] (B) Fault Recovery

[53] Database synchronization should be maintained even when a fault is generated in one system of the duplex systems. For example, suppose a fault is generated in the remote system (secondary system) 200. The recovery operation for synchronizing the remote system 200 with the local system 100 will be described.

[54] Figure 7 is a block diagram showing a preferred embodiment of an apparatus for achieving database synchronization between the system having the fault and the normal system. Processes b through g and i designate the recovery process of the normal system that are performed, processes b' through f' designate the ASE operation of the normal system, and process h designates the operation of the faulty after it is restarted.

[55] Figures 8A and 8B are flow charts showing a preferred embodiment method for recovering the system, in which a fault is generated, so that database synchronization with the normal system can be achieved. For the sake of convenience, the system 200 having the fault is referred to as the Fault System (FS) and the normally operated system 100 is referred to as the Normal System (NS). FS 200 is restarted for its recovery by the operator (S61 and S62). The PRMB 240 of the restarted FS 200 transmits a recovery completion message to the recovery process 134 and to ASE 131 of the NS 100 (S63) (b and b'). The recovery process 134 is driven when a fault is generated in one system between the duplex systems. That is, it is driven when it receives the recovery completion message from the FS 200 and is not operated when the duplex systems are normally operated. ASE 131 is operated when the systems are normally operated and when a fault is generated in one system.

[56] When the recovery process 134 of the NS 100 receives the recovery completion message, it starts a sequential search within the CLT 112 (c). ASE 131 processes messages received from the counterpart system 300, since the fault was generated in the FS 200 (c') (S64). Information about a transaction requiring synchronization that is performed in one system and is not reflected to the other system is recorded in CLT 112. Therefore, if a transaction requiring synchronization is performed after the fault is generated in the remote system 200, the ASE 131 of the local system 100 records the performed transaction to the CLT 112 (d'). Also, if the transaction list is found in the CLT 112, the recovery process 134 sequentially searching the CLT 112 selects the corresponding transaction from the UPT 111 using a key of the found

transaction (d) (S65). The recovery system does not select the update contents currently being processed in the UPT 111 by ASE 131.

[57] ASE 131 and the recovery process 134 of the local system 100 reflect the transaction requiring the synchronization, which was identified during the search within the CLT 112 and recovered from the UPT 111, to the remote system 200 (e and e') (S66). If the reflection of the update contents to the UPT 211 of the remote system 200 is successful (S67), the local system deletes the corresponding transaction list in the CLT 112 (f and f') (S68). However, if the reflection to the UPT 211 of the remote system is not successful, the local system 100 maintains the corresponding transaction list in the CLT 112 (S69). Recovery process 134 of the local system 100 identifies whether the sequential search in the CLT 112 is finished or not (S70), and if the search is not finished, the recovery process repeatedly performs processes S64 through S70.

[58] Therefore, database synchronization according to the present invention is not performed in the same way as that of the related art system. The related art system copies the entire database from an operational system to a restored system, after a fault occurs. In a preferred embodiment of the present invention, database synchronization between the duplex systems is achieved and maintained before the fault is generated and, therefore, only the data updated after the fault is reflected to the restarted system.

[59] After the sequential search of the CLT 112 is finished in process S70, the recovery process 134 of the local system 100 identifies whether the remote system is able to process messages (S71). If the remote system is able to process messages, the recovery process is ended.

If the remote system is not able to process messages, recovery process 134 transmits a message of inquiring message process, which informs the PRMB 240 that the database synchronization between the remote system 200 and the local system 100 is complete and the remote system 200 may start processing messages (g) (S72). The PRMB 240 of the remote system 200 transmits a message of inquiring message transmission to the counterpart system 300 so that the latter system will forward a message to be processed (h) (S73). Then, PRMB 240 of the remote system 200 transmits a message of starting message process to the recovery process 134 of the local system 100, as a response to the message of inquiring message process (i) (S74).

[60] Therefore, transactions requiring synchronization that were performed in the normal system, after the fault was generated, are reflected to the database of the faulty system after it is restarted. Thereby database synchronization is maintained.

[61] When the duplex systems are operated normally and a transaction requiring synchronization is performed in one system, if a corresponding transaction is in the CLT, the corresponding transaction is reflected to the other system in accordance with the type of current transaction operation and the type of transaction operation listed in the CLT. In this way, database synchronization is maintained in real-time in the duplex system. In addition, transactions requiring synchronization performed in one system and not reflected to the other system are reduced, thereby increasing the accuracy of database synchronization.

[62] When a fault is generated in one system of the duplex systems, the transactions requiring synchronization performed in the normal system, after the fault was generated, are reflected to the restarted system using the CLT of the normal system. There is no need to

initialize the database of the restarted system as in the related art. Also, the problem of the related art that the data transmission amount is increased because all the data of the normal system is transmitted to the initialized database can be solved. Moreover, the method according to the preferred embodiment of the present invention maintains database synchronization with reference to the CLT of the database, so access time to the disk and the data transmission time can be reduced.

[63] The foregoing embodiments and advantages are merely exemplary and are not to be construed as limiting the present invention. The present teaching can be readily applied to other types of apparatuses. The description of the present invention is intended to be illustrative, and not to limit the scope of the claims. Many alternatives, modifications, and variations will be apparent to those skilled in the art. In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents but also equivalent structures.